CLAIMS

1. A method for asynchronous brokering of messages between middleware computing systems, comprising:

a) receiving a message from a sending middleware computing system into a middleware brokering server; and

b) sending the message from the middleware brokering server to at least one receiving middleware computing system.

2. The method of claim 1 wherein the sending middleware computing system and the receiving middleware computing system are selected from the group consisting of a mainframe system, a CORBA compliant system, and a JMS system.

3. The method of claim 2 wherein the sending middleware computing system communicates with the middleware brokering server via point to point messaging and wherein the middleware brokering server communicates with the receiving middleware computing system via publish and subscribe messaging.

4. The method of claim 3 wherein the sending middleware computing system of a first message is the receiving middleware computing system of a second message.

5. The method of claim 3 wherein the receiving middleware computing system of a first message is the sending middleware computing system of a second message.

6. The method of claim 1 wherein the message is converted from a native language format of the sending middleware computing system to a structured event message format prior to being sent by the middleware brokering server.

7. The method of claim 6 wherein the message is converted from the native language format by mapping a plurality of fields in the native format into corresponding fields in the structured event message format.

8. The method of claim 7 wherein the native message format is a selected from the group consisting of a Cobol copybook, JMS TextMessage, JMS BytesMessage; JMS MapMessage; JMS ObjectMessage; and JMS StreamMessage.

9. The method of claim 7 wherein the sending middleware computing system is a mainframe system and the native message format is a COBOL copybook.

10. The method of claim 7 wherein the sending middleware computing system is a JMS system and the native message format is a JMS MapMessage.

11. The method of claim 1 wherein the message is converted from a structured event message format to a native language format of the receiving middleware computing system prior to being received by the receiving middleware computing system.

12. The method of claim 6 wherein the message is converted from a structured event message format to a native language format of the receiving middleware computing system prior to being received by the receiving middleware computing system.

13. The method of claim 7 wherein the message is converted from a structured event message format to a native language format of the receiving middleware computing system prior to being received by the receiving middleware computing system.

14. The method of claim 11 wherein the message is converted from the structured event message format by mapping a plurality of fields in the structured event format into corresponding fields in the native language format.

15. The method of claim 12 wherein the message is converted from the structured event message format by mapping a plurality of fields in the structured event format into corresponding fields in the native language format.

16. The method of claim 13 wherein the message is converted from the structured event message format by mapping a plurality of fields in the structured event format into corresponding fields in the native language format.

17. The method of claim 16 wherein the native message format is a selected from the group consisting of a Cobol copybook, JMS TextMessage, JMS BytesMessage; JMS MapMessage; JMS ObjectMessage; and JMS StreamMessage.

18. The method of claim 16 wherein the sending middleware computing system is a mainframe system and the native message format is a COBOL copybook.

19. The method of claim 16 wherein the sending middleware computing system is a JMS system and the native message format is a JMS MapMessage.

20. The method of claim 3 wherein the publish and subscribe messaging further comprises a push-pull paradigm across at least one messaging channel.

21. The method of claim 20 further comprising designating quality of service attributes when configuring the channel.